

Optimization-Oriented RAW Modeling of IEEE 802.11ah Heterogeneous Networks

Le Tian*, Elena Lopez-Aguilera [†], Eduard Garcia-Villegas [†],
Michael Mehari[‡], Eli De Poorter[‡], Steven Latré*, Jeroen Famaey*

*University of Antwerp - imec, IDLab, Department of Mathematics and Computer Science, Belgium

[†]Universitat Politècnica de Catalunya, Department of Network Engineering, Barcelona, Spain

[‡]Ghent University - imec, IDLab, Department of Information Technology, Belgium

Abstract—The new medium access method of IEEE 802.11ah, called Restricted Access Window (RAW), divides stations into different groups, and only allows stations in the same group to access the channel simultaneously, in order to reduce collisions and thus achieve better performance (e.g., throughput). However, the existing station grouping strategies only support homogeneous scenarios where all stations use the same modulation and coding scheme (MCS) and packet size. A surrogate model is an efficient mathematical model that represents the behavior of a complex system, trained with a limited set of labeled input-output data samples. In this paper, we present a surrogate model that can accurately predict RAW performance under a given Restricted Access Window (RAW) configuration in heterogeneous networks. Different from the homogeneous scenario, heterogeneous networks are defined by a large number of parameters, leading to an enormous design space, i.e., the order of 10^{17} possible data points. This is too big to achieve feasible training convergence. In this paper, we present a novel training methodology that leads to a new design space with highly reduced size, i.e., the order of 10^5 data points. The surrogate model converges when less than 6000 labeled data points are used for training, which is only a tiny portion of the whole design space. The results show that, the relative error between model prediction and simulation results is less than 0.1 for 95% of the data points, in the areas of the design space studied. Its low complexity and high precision make the proposed model a valuable tool to develop real-time RAW optimization algorithms for heterogeneous IEEE 802.11ah networks.

Index Terms—IEEE 802.11ah, RAW, surrogate model, heterogeneous networks.

I. INTRODUCTION

The new Wi-Fi standard IEEE 802.11ah [1], targeting large scale and low-power Internet of Things (IoT) network scenarios, proposes a novel station-grouping based medium access method, referred to as Restricted Access Window (RAW). In particular, stations are divided into groups, and a group of stations are only allowed to access the channel during a specific time slot, in order to reduce contention and collisions in highly dense deployments. It is a flexible hybrid channel access method, allowing up to 8192 stations connected to a single Access Point (AP), highly suitable to provide scalable connectivity to both sparsely and densely deployed low-power devices.

Copyright (c) 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

Figure 1 schematically depicts how RAW works. Specifically, the airtime is split into intervals, some of which are assigned to RAW groups, while others are considered as shared channel airtime and can be accessed by all stations. A beacon frame is transmitted periodically, carrying a RAW Parameter Set (RPS) information element. The RPS specifies the stations that belong to each RAW group, as well as the interval start time. Moreover, each RAW interval consists of one or more slots, over which the stations in the RAW group are split evenly using round robin assignment. Therefore, the RAW related parameters include the number of stations assigned to each RAW group, number of RAW groups, duration of each RAW group, and number of slots in each RAW group.

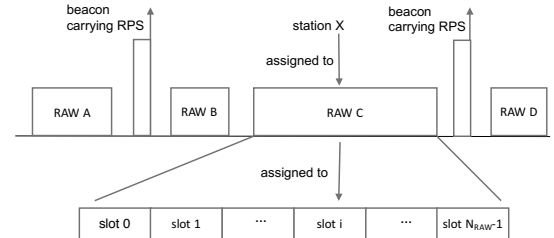


Fig. 1: Schematic representation of the RAW mechanism.

An extensive evaluation on RAW has been conducted in [2], the results therein demonstrate that, with appropriate RAW configuration, the RAW mechanism can provide substantial performance improvement (e.g., throughput, energy consumption) over the legacy channel access method Enhanced Distributed Channel Access and Distributed Coordination Function (EDCA/DCF), particularly for highly-loaded dense network scenarios. On the contrary, the incorrect RAW configuration severely deteriorates performance. Moreover, it reveals the optimal RAW parameters are affected by a variety of network-related parameters, e.g., the number of stations, traffic patterns, and network load. Therefore, in order to optimally configure RAW, a model is needed, capable of predicting the performance for the given RAW parameter values and network conditions. Concretely, the model takes network conditions and a RAW configuration as input, and predicts one or more performance metrics (e.g., throughput or energy consumption) as output. Although several RAW models have been proposed in literature [3], [4], [5], [6], [7], [8], they are either too computationally hard to be used in real-time,

or rely on certain simplifications (e.g., no capture effect, no communication errors, homogeneous stations, saturated traffic, etc.). Such simplifications limit the usage of the RAW models, making it difficult to apply them to realistic IoT networks.

In this paper, we present a surrogate model for RAW performance in realistic IoT scenarios. The model is trained using realistic simulation results (with capture effect enabled and the presence of communication errors). It can accurately predict the performance under a given RAW configuration in IEEE 802.11ah heterogeneous networks, in which stations are allowed to use different packet sizes based on the application requirements and MCSs based on their distance to the AP. To build such a model, based on our knowledge on IEEE 802.11ah and surrogate modeling, a novel training methodology is proposed, including each aspect of the modeling process, i.e., the design of input and output parameters, initial design points selection, surrogate model creation, sampling strategies and stopping criteria. Among them, the design of input and output parameters is most challenging and critical. The heterogeneity of the network and the RAW configuration is represented by many input variables, leading to a design space that is too huge to train. However, our proposed method significantly reduces the design space, allowing the model to be accurately trained with very few labeled sample data points. Moreover, once trained, evaluating the model is equivalent to a constant-time table lookup, which can be easily executed in real-time. To the best of our knowledge, this is the first RAW model that supports heterogeneous stations in terms of different modulation and coding schemes (MCSs) and packet sizes. The built model can be used as an input for real-time optimization algorithms such as [9], [10], [11], [12] to get optimal RAW configurations for IEEE 802.11ah heterogeneous networks, achieving highest performance.

The remainder of this paper is structured as follows. Section II describes the IEEE 802.11ah and its RAW feature, surveys the related work on RAW performance modeling and compares it to our contribution. The training methodology of RAW surrogate model is described in Section III. Section IV evaluates the performance of our presented model. Finally, Section V offers conclusions and a short overview of future work.

II. BACKGROUND

A. IEEE 802.11 ah networks and RAW

In this section, we provide an overview of RAW, the IEEE 802.11ah standard and its new PHY and MAC layer characteristics.

IEEE 802.11ah mainly targets IoT network scenarios, where sensors (stations) are placed around the AP, periodically monitor the environment and send the resulting data to a server (via the AP). At the PHY layer, IEEE 802.11ah supports multiple transmission data rates represented by the MCSs. The stations are allowed to dynamically choose the MCS in order to adapt to the conditions of the wireless channel. As listed in Table I, the supported MCSs depends on the channel width. For channel bandwidth of 1 MHz, MCS 0 to 10 are supported with transmission data rates ranging from 150 Kbps

TABLE I: 802.11ah MCSs for 1, 2 MHz, NSS=1, GI=8 μ s.

MCS Index	Modulation	Coding rate	Data rate (Kbps)	
			1 MHz	2 MHz
0	BPSK	1/2	300	650
1	QPSK	1/2	600	1300
2	QPSK	3/4	900	1950
3	16-QAM	1/2	1200	2600
4	16-QAM	3/4	1800	3900
5	64-QAM	2/3	2400	5200
6	64-QAM	3/4	2700	5850
7	64-QAM	5/6	3000	6500
8	256-QAM	3/4	3600	7800
9	256-QAM	5/6	4000	Not valid
10	BPSK	1/2 with 2x repetition	150	Not valid

to 4 Mbps, and for 2 MHz, MCS 0 to 9 are supported with transmission data rates ranging from 650 Kbps to 7.8 Mbps. Besides, IEEE 802.11ah also supports channel bandwidth 4, 8 and 16 MHz, more details can be found in [1]. At the start of every beacon interval, the AP broadcasts a beacon frame carrying an RPS information element specifying the RAW parameter configurations. Stations retrieve such RAW information from the beacon frame and access the channel only during their assigned RAW slot.

The RAW information carried in the RPS element specifies the stations belonging to the group, the number of slots, slot format and slot duration count sub-fields, which jointly determine the RAW slot duration (D) as follows [1]:

$$D = 500 \mu s + C \times 120 \mu s \quad (1)$$

where C represents the *slot duration count* sub-field, which is either $y = 11$ or $y = 8$ bits long if the slot format sub-field is set to 1 or 0, respectively. The *number of slots* field is $14 - y$ bits long. For $y = 11$, each RAW consists of at most 8 slots and the maximum value of C is $2^{11} - 1 = 2047$. In this case the slot duration is up to 246.14 ms. Otherwise, each RAW consists of at most 64 slots and the maximum value of C is $2^8 - 1 = 255$, the slot duration is thus limited to 31.1 ms. The RAW group duration is the sum of its corresponding slot durations. For a more in-depth description, the reader is referred to existing literature [1], [13], [14].

The RPS also contains the cross slot boundary (CSB) field. Stations are allowed to continue ongoing transmissions after the end of the current RAW slot when CSB is set to true. Otherwise, stations should not start a transmission if the remaining time in the current RAW slot is not enough to complete it.

B. Surrogate modeling

A surrogate model [15] is an efficient mathematical model that represents the behavior of a complex system such as a circuit, a flight motion or, in this case, a wireless network. A surrogate model is trained at design time, using a limited number of labeled input-output sample data points obtained through simulation (e.g., ns-3) or real-life experiments. As illustrated in Figure 2, surrogate modeling mainly involves four parts:

- **Initial design:** Select initial data points as a first simple design space representation, then build an initial model with the selected initial design points by applying supervised machine learning and regression methods.
- **Sampling strategy:** Select a few new sample data points to improve the model if the model's accuracy is not satisfactory.
- **Model creation:** Building an intermediary model together with the existing and newly selected data points.
- **Stopping criteria:** Define the stopping criteria, stop the training process once the stopping criteria are satisfied, otherwise, select the new sample data points and update the model.

As the IEEE 802.11ah standard does not specify how to configure the actual RAW grouping parameters, several studies have been conducted on RAW performance evaluation, modeling and optimization. Raesi *et al.* evaluated RAW performance using OMNeT++, the results demonstrate that the RAW feature provides substantial performance improvements (e.g., throughput, energy consumption), particularly in scenarios where there are high number of collisions in a heavily loaded network [4]. Using an IEEE 802.11ah implementation in ns-3 [19], we further evaluated the optimal RAW configuration under a variety of network conditions, such as traffic load, number of stations, and traffic distribution [2]. These works highlight the impact of network conditions on the optimal RAW configuration, and demonstrate there is a need for real-time RAW optimization under dynamic network conditions. To accomplish such an optimization algorithm, a RAW performance model is required. The model is expected to be able

The above mathematical models have two main disadvantages. First of all, some of them are computationally hard. This makes it infeasible to execute them in real-time on actual AP hardware, where at most a few milliseconds are available at the start of the beacon interval to calculate a new RAW configuration. More importantly, they all assume certain types of traffic and ideal channel conditions, without communication

errors, delays or capture effects. The combination of these factors make such models useful only from a theoretical point of view, to analyze the effectiveness of RAW under a variety of conditions. However, their utility is limited for real-time station grouping under dynamic and realistic traffic conditions.

Chang *et al.* went a step further to support more diverse traffic demands [21]. They used the results of two extreme cases (i.e., with infinite traffic and with only a single packet) to extrapolate a regression-based analytical model that can accurately fit the contention success probability of any traffic patterns. However, the model does not take the finite length of the RAW slot into account. Recently, we proposed a new RAW performance model based on supervised surrogate modeling [11], [12]. The model is trained on a limited set of labeled data samples from ns-3 simulation results, supports realistic channel conditions, including communication errors, propagation delays and capture effects. However, the model only supports homogeneous networks, i.e., all stations in the same network use the same MCS and packet size.

In this paper, we present a novel training methodology for surrogate modeling of IEEE 802.11ah heterogeneous networks, in which stations can have different MCSs and packet sizes based on the services they are targeting. It uses average transmission time (i.e., the total packet transmission time of a heterogeneous network averaged over all the packets sent) as an input parameter of the surrogate model, and packet receiving rate (i.e., number of packets received per second) as the output parameter. This results in a highly reduced design space, allowing the model to be accurately trained with only a few labelled sample data points.

III. SURROGATE MODEL OF IEEE 802.11AH RAW

This section first describes IEEE 802.11ah heterogeneous networks used for training. Subsequently, the input and output parameters of the surrogate RAW model are designed, and training steps for the IEEE 802.11ah heterogeneous networks are described.

A. Training scenarios

For the training, we use the common parameter settings, as shown in Table II. Given the low-power nature of battery powered sensors, the PHY layer parameters are configured based on the low-power IEEE 802.11ah radio hardware prototype developed by Ba et al. [22], with a transmission power of 0 dBm, a gain of 0 dB (for both station and AP), and noise figure of 6.8 dB. With such common PHY settings, the coverage of the network is up to 500 meters based on the physical layer performance evaluation in [23]. As IoT applications use a relatively small payload size, packet size is assumed to be between 32 and 512 bytes. For each evaluated network, we assume a coverage range to be $[d^-, d^+] \subseteq [0, 500]$ meters and a packet size range to be $[ps^-, ps^+] \subseteq [32, 512]$ bytes. Each station s randomly and uniformly chooses a value $d_s \in [d^-, d^+]$ as its distance to the AP, and an integer value $ps_s \in [ps^-, ps^+]$ as its packet size. The station s uses the corresponding MCS for packet transmission according to the rate control method. Several rate control methods have been

TABLE II: Simulation parameters used during training

PHY parameters	Value
TX power	0 dBm
TX/RX gain	0 dB
Noise Figure	6.8 dB
Propagation model	macro [23]
Error rate model	YansErrorRate
MAC parameters	Value
Traffic access categories	AC_BE
Beacon interval	204.8 ms
Size of transmission queue	10 packets
Packet transmission interval	1 s
Station distribution	Random

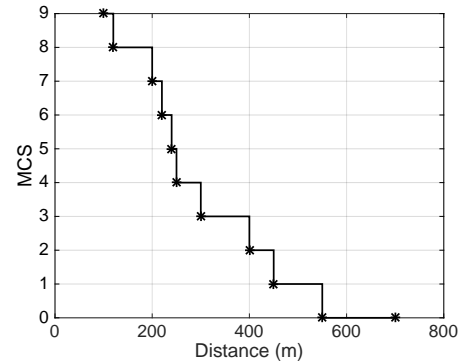


Fig. 3: MCS as a function of the distance between a station and AP for 1 MHz bandwidth.

proposed and used on real devices for legacy IEEE 802.11 to select the appropriate MCS for packet transmission, for example, Arf [24], Aarf [25], Onoe [26] and Minstrel [27]. In this paper, we aim to model instantaneous throughput of a single RAW group (i.e., one beacon interval at most), MCS changes at such a short time interval are not expected. As such, we allow the stations to select the MCS solely based on their distance d_s from the AP, i.e., choosing the MCS which can stably achieve the maximal throughput at a certain distance, as shown in Figure 3. For training simplicity, we assume each station sends one packet per second and a small buffer size of 10 packets is used. The built model can be further used by the RAW optimization algorithms, such as TAROA [9], [10] and MoROA [11], [12], to calculate RAW performance under arbitrary data transmission intervals.

B. Parameter design for surrogate RAW modeling

The goal is to build a surrogate model with a limited number of sample input-output data points, which can accurately predict the performance of a heterogeneous IEEE 802.11ah network for a given RAW configuration and network condition. Therefore, a set of parameters should be defined to represent the network conditions and RAW configurations. Moreover, the input parameter space needs to be defined, which consists of the minimum and maximum value of each parameter, as well as a step size. The size of the input parameter space represents the number of data points of the surrogate model.

TABLE III: Definition of the input parameter space.

Parameter	Description	Min	Step	Max
n_r	Number of stations	60	5	400
d_r (μs)	Duration of RAW group	40960	5120	204800
s_r *	Number of RAW slots	1	5	50
\hat{t}_p (μs)	Average transmission time	1000	250	5000

* The values of s_r is 1, 5 and all multiples of 5 until 50.

Therefore, the surrogate model of IEEE 802.11ah RAW can be generalized as follows:

$$o = \mathcal{F}(I_r, I_n) \quad (2)$$

where I_r represents a set of RAW-related input parameters, I_n is a set of network-related input parameters, and the performance metric is represented by o . The parameter o should be able to represent the performance metrics (e.g., throughput, energy consumption or latency) with the given input parameters. To simplify the explanation, we focus on throughput as the output metric, but a similar method can be used to train a model for energy consumption or latency. The size of the design space is defined as follows:

$$\mathcal{C} = c_r \times c_n = \prod_{i_r \in I_r} |i_r| \times \prod_{i_n \in I_n} |i_n| \quad (3)$$

where $|i|$ represents the total number of possible values of input variable i , c_r and c_n represent the total number of values that can be taken by RAW-related parameters I_r and network-related parameters I_n , respectively. Among the data points inside the design space, a few of them are selected as sample data points to train the surrogate model, in order to accurately predict the output of the other data points. Therefore, the parameters and their step sizes should be well chosen, leveraging the trade-off between accuracy and training speed. On the one hand, a small number of parameters and large step sizes lead to a small design space, which needs less training time but at the cost of losing certain accuracy. On the other hand, a large number of parameters and small step size result in a large design space, which can more accurately characterize the system but needs more training time.

1) RAW parameters I_r design

As mentioned in Section II-A, the related parameters of a RAW group r are: number of stations assigned to the RAW groups n_r , duration of the RAW group d_r , and number of slots in the RAW group s_r . In our previous work [11], we selected the appropriate range and step size for each RAW parameter, which leads to high model accuracy and only a limited number of training data points. Therefore, in this paper, we use the same RAW parameter values, as listed in Table III. This highly reduces the design space size of RAW-related parameters from $c_r = 2.3 \times 10^7$ to $\tilde{c}_r = 25047$. It should be noted that, as the minimal value of s_r is 1 according to the IEEE 802.11ah specification, we use step size of 4 for $s_r = 1$ and the next value (i.e., $s_r = 5$), and step size of 5 for other s_r values. For more details, the readers are referred to our previous work [11].

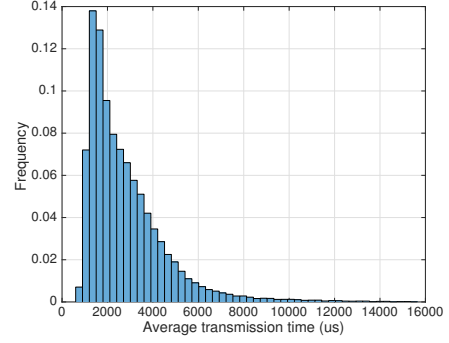


Fig. 4: Distribution of average transmission time of IEEE 802.11ah networks supporting different coverage and packet size ranges in the described scenario.

2) Network condition parameter I_n design

As in the IEEE 802.11ah network environment, the common settings for an IoT scenario are represented by the parameters listed in Table II, the remaining variables are the packet size range $[ps^-, ps^+]$ and coverage range $[d^-, d^+]$ (determining the MCS), which leads to a variety of heterogeneous networks. In a naive approach, one would use parameters $\{ps^-, ps^+, d^-, d^+\}$ as the network-related input parameters I_n , representing the networks conditions. However, this would result in a huge input design space, significantly increasing the required training data points, and therefore slowing down the training speed. According to the training scenarios, d^- and d^+ are between 0 and 500 meters with $d^- \leq d^+$, ps^- and ps^+ are between 32 and 512 bytes with $ps^- \leq ps^+$. With step size 1, the coverage range results in c_d different combinations:

$$c_d = \sum_{d^-=0}^{512} |d^-| \quad (4)$$

The parameter $|d^-|$ represents the number of possible combinations caused by d^- and equals to $(512 - d^- + 1)$, leading to $c_d = 125751$ combinations. Similarly, the packet size range results in $c_{ps} = 115921$ combinations. In total, the coverage range and packet size range results in $c_n = c_d \times c_{ps} = 1.4 \times 10^{10}$ combinations considering different network condition parameters. Therefore, the total number of data points of the input space is $\mathcal{C} = c_r \times c_n = 3.2 \times 10^{17}$. With a large step size, for instance, 100 for the coverage and 120 for the packet size, the input space still has 7.2×10^9 data points. Both cases result in a design space that is too big to achieve feasible convergence. Therefore, as MCS and packet size jointly determine the transmission time of a packet, we use average packet transmission time T_x of all stations as the input parameter I_n representing network conditions, leveraging the trade-off between accuracy and training speed. We define average transmission time T_x , as the average airtime per packet transmission in the network. The remainder of this section elaborates the highly reduced input design space, including the distribution of average transmission time T_x among different IEEE 802.11ah network scenarios, and its minimum, maximum and step values. Section III-D provides a detailed discussion on the accuracy of the model.

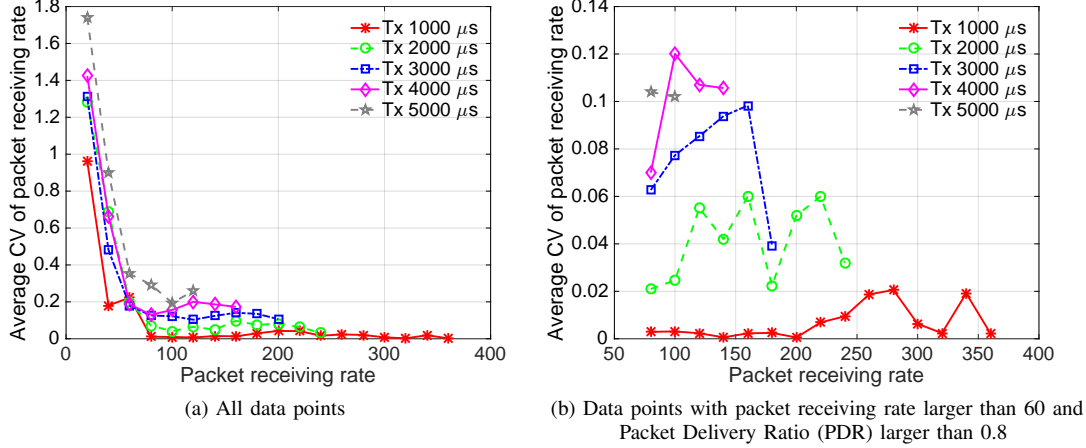


Fig. 5: Average CV of packet receiving rate as a function of packet receiving rate for different average transmission time.

Figure 4 depicts the distribution of average transmission time T_x for the training scenarios. The figure is derived from the simulation results of a broad range of networks which have different coverage and packet size ranges. The d^- and d^+ parameters are selected from $[min:step:max]=[0:10:500]$ meters, and the ps^- and ps^+ belong to $[min:step:max]=[32:32:512]$ bytes. Therefore, the combination of coverage and packet size range leads to 180336 different IEEE 802.11ah networks. We calculate the average packet transmission time across all stations in each network, and count how many times a certain average transmission time occurs among the 180336 simulated scenarios. The results in Figure 4 show that the average transmission time in the range of $[1200\ 1500]\ \mu s$ is the most frequent (13.7%, 24660 times), and the average transmission time is mainly between 1000 and 5000 μs (around 90% of all experiments). Therefore, when average transmission time is considered as an input parameter of the model, 1000 and 5000 μs can be used as its minimum and maximum values for the training. A step size of 250 μs is found a good choice to leverage the trade-off between accuracy and training speed, leading to $\tilde{c}_n = 17$ different values for average transmission time. Therefore, the utilization of average transmission time results in a design space with only $\tilde{C} = \tilde{c}_r \times \tilde{c}_n = 425799$ data points, from which samples are drawn during the training. It is only about one out of 7.5×10^{11} possible data points, compared to the design space of size $C = 3.2 \times 10^{17}$, where d^- , d^+ , ps^- and ps^+ are used as input parameters to represent the network.

3) Output parameter design

As an average transmission time can consist of a variety of packet size ranges that can affect the throughput, the packet receiving rate (i.e., number of received packets per second at the AP) is used as the metric of performance. The throughput can be subsequently calculated considering the average packet size for a specific network scenario.

Therefore, the surrogate model described in Eq 2 can be concretely represented as a function \mathcal{F} :

$$p_r = \mathcal{F}(n_r, d_r, s_r, T_x) \quad (5)$$

where p_r represents the packet receiving rate. It aims to predict the performance of a heterogeneous IEEE 802.11ah network with average packet transmission time T_x , for a given RAW group r with duration d_r , consisting of s_r slots, and with n_r stations assigned to it. The definition and evaluated ranges of the input parameters are listed in Table III.

C. Network condition parameter analysis

In this section, we demonstrate that the average transmission time T_x is an appropriate input parameter for the model, as it can accurately represent the network conditions for the optimization purpose with highly reduced design space size. As the size of the design space has been discussed in Section III-B2, this section focuses on the accurate representativeness. The accurate representativeness is used to show that, although the same average transmission time T_x can be produced in very different networks (i.e., having different MCS and packet size distributions), they show similar behavior in terms of packet receiving rate p_r , allowing to accurately predict one from the other. In this section, the Coefficient of Variation (CV) of the packet receiving rate p_r is used as the criterion to evaluate the representativeness of T_x . The CV, also known as relative standard deviation, is a measure of variation or dispersion of a set of data values. It is defined as follows:

$$CV = \frac{\sigma}{\mu} \quad (6)$$

where σ is the standard deviation of a series of values, and μ is the average value.

The variance is evaluated under different packet receiving rates p_r and RAW slot load ratios \mathcal{L}_r . The RAW slot load ratio \mathcal{L}_r is defined as the ratio between the required transmission time of the packets that are allowed to be transmitted in a RAW group, and the duration of the RAW group. We define it as follows:

$$\mathcal{L}_r = \frac{(T_x + T_{SIFS} + T_{ACK}) \times n_r}{d_r \times \mathcal{T}_s} \quad (7)$$

where \mathcal{T}_s represents the packet sending interval of a station. T_{SIFS} and T_{ACK} are the amount of time required

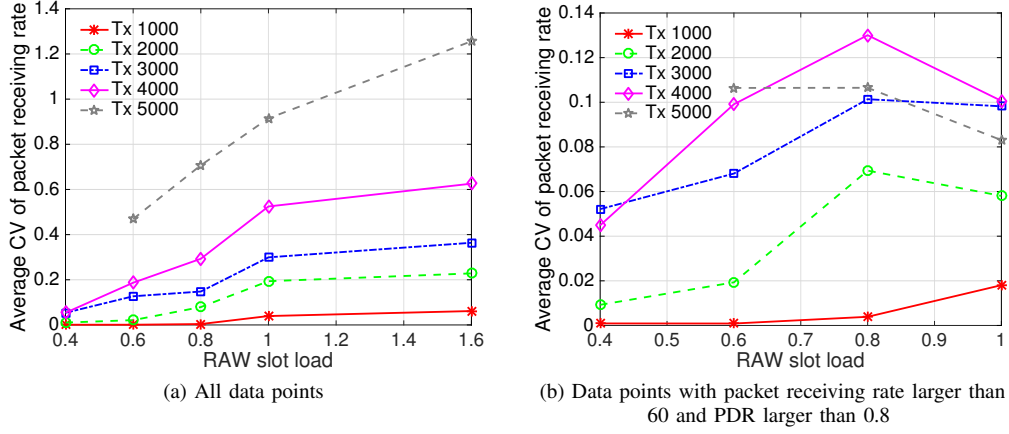


Fig. 6: Average CV of packet receiving rate as a function of RAW slot load rate for different average transmission time.

for transmitting SIFS and ACK frame, respectively. For IEEE 802.11ah, T_{SIFS} is 160 μ s, and T_{ACK} is 1000 μ s for 1 MHz bandwidth. A large value of L_r means a high input traffic load, and vice versa. The network is considered overloaded when the slot load ratio \mathcal{L}_r is larger than 1.

For the evaluation, 500 design points for each average transmission time T_x (i.e., 1000, 2000, 3000, 4000 and 5000 μ s) are randomly selected, consisting of 5 different RAW slot load ratios \mathcal{L}_r (i.e., 0.4, 0.6, 0.8, 1.0, 1.6). Each data point has four input parameters, i.e., number of stations n_r , number of slots s_r , RAW duration d_r , and average transmission time T_x . The simulation runs 10 times for each data point. During each simulation, the simulator randomly selects a different coverage and packet size range combination satisfying the required average transmission time. After all simulations, the CV of each data point's output (i.e., packet receiving rate) is calculated as depicted in Eq 6. Subsequently, for each average transmission time T_x , the CV of data points with the same packet receiving rate p_r and RAW slot load ratio \mathcal{L}_r is averaged, respectively. The final results are shown in Figure 5 and 6. Each depicts the results with all data points (subfigure a) and only the data points with packet receiving rate p_r larger than 60 Packets per Second (PPS) and PDR larger than 0.8 (subfigure b). PDR is defined as the ratio between the number of sent packets and the number of received packets.

Figure 5a shows that a smaller T_x results in less variability, while a larger average transmission time T_x leads to higher variability. Moreover, a small packet receiving rate p_r (i.e., less than 60) causes quite high performance discrepancy, with CV 0.96 and 1.74 for average transmission time T_x 1000 μ s and 5000 μ s, respectively. However, when focusing on packet receiving rate p_r larger than 60 and PDR larger than 0.8 (i.e., feasible and practical network conditions), the CV is significantly reduced, as depicted in Figure 5b. For example, the CV is at most 0.02, and between 0.02 and 0.06, for average transmission time T_x 1000 and 2000 μ s, respectively. For average transmission time T_x 5000 μ s, the CV ranges from 0.10 to 0.12, more than 10 times smaller than when considering all values of p_r and PDR. Figure 6a further reveals that the performance variation increases with larger RAW slot load \mathcal{L}_r . However, by applying the same constraint as in Figure

5b (i.e., practical network conditions with $p_r > 60$ and PDR > 0.8), the CV is significantly reduced, as depicted in Figure 6b. For example, when average transmission time is 1000 μ s, the CV is 0.001 and 0.02 for RAW slot load ratio \mathcal{L}_r 0.4 and 1.0, respectively. It should be noted that, the RAW slot load ratio \mathcal{L}_r is up to 1 in Figure 6b due to the constraint on PDR.

The fact that the performance variance is affected by different parameters, including packet receiving rate, RAW slot load ratio, PDR and average transmission time, is explained as follows.

- **Packet receiving rate p_r :** The large CV values appearing at small packet receiving rate are due to the fact that small values more easily lead to a large variance, as small absolute errors have a large relative effect.
- **RAW slot load ratio \mathcal{L}_r :** Networks have different input parameters, and even the networks sharing the same input parameters (i.e., average transmission time T_x) may have different coverage and packet size ranges. Thus, there are different saturated traffic levels for these networks. With a larger RAW slot load ratio, the channel contention is more fierce, leading to more networks reaching their saturated levels, and therefore, resulting in more discrepancy in their performance.
- **PDR:** Higher PDR implies that corresponding data points have reasonable RAW configurations, thus mitigating channel contention. Therefore, less performance discrepancy among different networks is obtained in this case.
- **Average transmission time T_x :** For each network with a given T_x , Figure 7 depicts its standard deviation of packet transmission time across all stations, as represented by a dot. It shows, for networks sharing a larger average transmission time, there is a higher chance that these networks have different packet transmission time distributions. Such difference can result in performance discrepancies when packet collisions happen, as the air-time wasted by collision and spent in re-transmissions depends on the transmission time of colliding packets.

The results shown in Figure 5 and 6 jointly reveal that average transmission time T_x can significantly reduce the design space (as illustrated in Section III-B2), but may add inaccuracies, since a given T_x can represent a number of

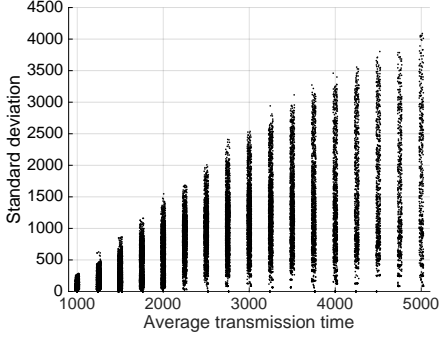


Fig. 7: Standard deviation of packet transmission times for networks with different average packet transmission time.

networks with slightly different behavior. However, the large performance variability concentrates in poor performing networks, including networks with small packet receiving rate p_r and low PDR. A low packet receiving rate signifies a low throughput, such scenarios have been shown not to be affected by the RAW configuration, and even perform better without RAW [2]. A low PDR signifies high packet loss, which means the network is heavily overloaded. As the model is built mainly for optimization purpose, we can still claim that the average transmission time T_x parameter is able to represent the network conditions in the most relevant scenarios for RAW optimization. Moreover, the accuracy increases with lower T_x and \mathcal{L}_r .

D. Training methodology

The training of the RAW performance surrogate model follows the steps listed below, as shown in Figure 2:

- 1) Based on the defined design space (cf. Table III), the initial design points are carefully selected to efficiently characterize the system. Each data point has four input parameters, i.e., number of stations n_r , number of slots s_r , RAW duration d_r , and average transmission time T_x .
- 2) The ns-3 simulator executes experiments with parameters from each initial data point and the general parameters of IEEE 802.11ah networks (cf. Table II). The evaluation criterion (i.e., packet receiving rate) of the experiments is considered as the output of each data point.
- 3) After the experiments with all the initial data points, an initial surrogate model is created.
- 4) The sampling strategy is applied to select the new data points to improve the model accuracy.
- 5) The experiments are executed for the newly selected data points, the evaluation criterion of the experiments is considered as the output of the new data points.
- 6) The surrogate model is updated with the newly selected data points.
- 7) This process stops once the stop conditions are met, otherwise it continues by repeating from step 4.

Each experiment in ns-3 runs for 60 seconds of simulated time. As RAW is configured in each beacon interval of 204.8 ms, the results of every simulated configuration are

averaged over 290 beacon intervals, ensuring the generality of the trained model. The proper method is applied in each step in order to create an accurate surrogate model with a limited number of data points. More details can be found below.

- **Initial design points selection**

We use the latin hypercube design (LHD) [28] approach to select the initial data points, as it has the best performance in general and is widely used. It selects sample points evenly along the configuration space, while ensuring proportional representation of design variables. Furthermore, the initial sample size depends on the problem type, 200 initial data points were found a good choice for our problem.

- **The (initial) surrogate model creation**

There are a variety of supervised machine learning and regression methods that can be used for surrogate model creation, such as Kriging [29], polynomial response surfaces [30], radial basis functions [31], support vector machines (SVMs) [32], space mapping [33], or artificial neural networks (ANNs) [34]. Among these methods, Kriging model is very popular to model complex systems, including complex wireless networks [16], [11], [18]. The Kriging model is formed as:

$$\hat{f}(X) = \sum_{i=1}^V a_i k(x, x_i) \quad (8)$$

where V is the amount of basis data vectors, x represents the input data vector and $k(x, x_i)$ is the kernel function. There are several kernel (covariance) functions used in Kriging, such as Squared Exponential Kernel, Exponential Kernel, Matern 3/2, Matern 5/2, Rational Quadratic Kernel. Among them, the Matern types of kernel functions are widely used. In our case, the Matern kernel function with 5/2 was found a good choice to create the model.

- **Sampling strategies**

A novel sampling strategy called FLOLA-Voronoi [35] is used to select the next design points in order to improve the model accuracy. The FLOLA approach is used for exploiting the non-linear regions, while the Voronoi approach explores the sparsely sampled regions. The scores from FLOLA and Voronoi are combined to decide the next sample point. In our experiments, 10 new data points are picked in each iteration.

- **Stopping Criteria**

The 10-fold cross-validation with a Root-Relative Square Error (RRSE) measure is used to evaluate the model accuracy. The modeling process stops once the cross validation score remains stable (3 digits of precision) for 10 successive iterations, i.e., 100 sample data points. Moreover, we set an upper limit of 10000 sample data points. The execution stops once the iteration number reaches the limit, even if the accuracy condition of 3 digits of precision is not met.

IV. PERFORMANCE EVALUATION AND DISCUSSION

This section presents the evaluation results on the RAW performance predication's accuracy of our surrogate model.

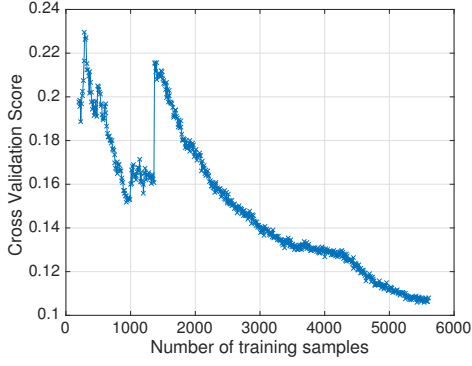


Fig. 8: Cross validation score of the surrogate model as a function of the number of training samples.

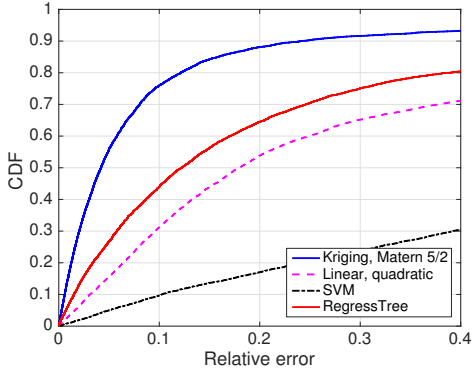


Fig. 9: Performance comparison between different regression models and simulation results for the reduced design space.

First, the convergence of the training process is evaluated. Subsequently, the accuracy of the surrogate model for the reduced design space and for the full design space are discussed respectively. As mentioned in Section III-B, there is a number of coverage and packet size ranges that can lead to the same average transmission time. For the reduced design space, the average transmission time is only represented by distance and packet size range combinations used during the training process. While in the full design space, there is no such limitation.

Our previously developed IEEE 802.11ah ns-3 simulation module [19], [36] is integrated into the Matlab Surrogate Modeling (SUMO) toolbox to train the RAW model. The SUMO Toolbox is a flexible framework for accurate global surrogate modeling [15], and has already been applied successfully to a wide range of applications. The behavior of each software component is configurable, and the components can easily be added, removed or replaced by custom implementations. Both training and evaluation use the IEEE 802.11ah scenario as described in Section III-A.

A. Training convergence

In this section, we evaluate the training convergence of the surrogate model. The cross validation score is widely used to measure the accuracy of the resulting model, a low score signifies a high accuracy of the model, and vice versa. Figure 8

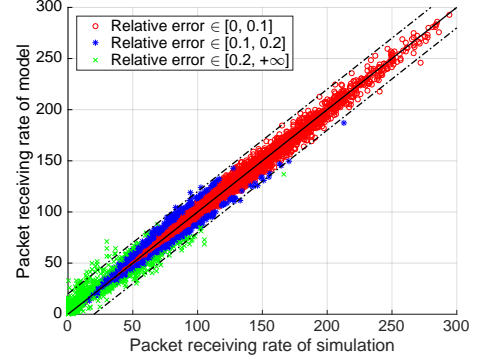


Fig. 10: Surrogate model prediction vs. simulation results.

demonstrates the 10-fold cross validation score of the model as a function of the number of training samples used. It clearly shows that, at first, SUMO is trying to learn the behavior of the networks with cross validation score going up and down. It should be noted that there are big jumps around the 1000th and 1360th sample data points, indicating that the SUMO toolbox discovers an untouched area that has a huge performance difference compared to earlier explored areas.

However, from 1360 samples onward, the cross validation score continually decreases until 5520 samples used for training, reaching 0.107. Subsequently, the cross validation score remains almost constant, signifying that the training process has converged. Training of the model stops after 5610 samples as the stop condition is satisfied, i.e., 10 consecutive cross-validation scores lower than or equal to 0.107 (3 digits of precision) are obtained. This comes down to about 1.3% of the reduced parameter space (i.e., 425799) defined in Table III, from which samples were drawn during the training process. It is only about one out of 5.7×10^{13} data points in the full design space.

B. Reduced design space experiments

In this section, we evaluate the RAW model accuracy for the reduced design space, in which the average transmission time is obtained only from the coverage and packet size ranges used during the training process. In this case, 6000 data points (called test data points) are randomly chosen. To evaluate the accuracy of the Kriging model (with Matern 5/2) used in the model creation step, three additional regression models (i.e., linear regression, Support Vector Machine (SVM) and regression trees) are also created with the trained data points for comparison. The simulation results of the 6000 test data points are compared to the prediction of the above models.

Figure 9 depicts the Cumulative Distribution Function (CDF) of the relative error, i.e., the ratio between the absolute error of the simulation results S_r with respect to the model prediction P_r , and simulation results S_r , i.e.,

$$\text{Relative error} = \frac{|S_r - P_r|}{S_r} \quad (9)$$

It shows 76% of the test data points have a relative error equal to or less than 0.1 for the Kriging model with Matern 5/2,

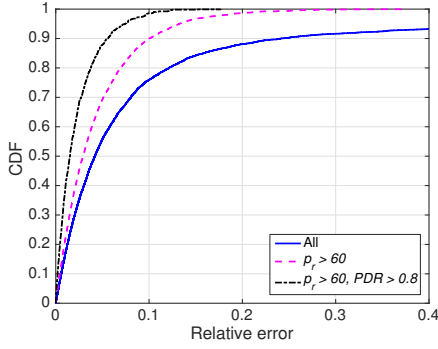


Fig. 11: Accuracy of surrogate model using Kriging method for different constraints in the reduced design space.

while 30%, 10% and 44% is obtained for the linear regression, SVM and regression tree models, respectively. Similarly, for the Kriging model, 88% of the test data points result in a relative error not larger than 0.2, while the percentage declines to 53%, 17% and 65% for the linear regression, SVM and regression tree models, respectively. Thus, the results show the Kriging model has higher accuracy than the other three regression models. However, there are still some data points with high relative error and further investigation is needed. It should be noted that the results in the reminder of this paper only consider the surrogate model created using the Kriging model.

To further explore the performance of the built surrogate model using Kriging, Figure 10 shows a scatterplot of the output of simulation and the predication of the model. The line $y = x$ represents the ideal case where the model can precisely predict the output of all data points without any errors. The points are color-coded based on the relative errors, with red meaning the relative error is lower than 0.1, blue representing the error between 0.1 and 0.2, and green indicating an error larger than 0.2. Figure 10 reveals interesting characteristics of the created RAW model, as the large relative errors mainly exist for the small packet receiving rate (i.e., in poorly configured networks), which makes sense as an estimation error for small packet receiving rate easily leads to a large relative error. Figure 11 depicts the CDF of the relative error for the Kriging model with packet receiving rate larger than 60 and PDR larger than 0.8. It shows, for packet receiving rate larger than 60, about 90% of the data points have a relative error equal to or less than 0.1. When further filtered with PDR larger than 0.8, about 98% of the data points show a relative error less than 0.1, and 88% for an relative error less than 0.05.

In conclusion, based on the above results, we discover that, for the reduced design space, the surrogate RAW model can accurately predict the output for data points with medium to high packet receiving rate (i.e., larger than 60). Moreover, the accuracy can be further improved when focusing on data points with high PDR (i.e., larger than 0.8). In the next section, the model accuracy for the full design space is evaluated.

C. Full design space experiments

In this section, we further explore the performance of the model for the full design space. In the full design space, a data

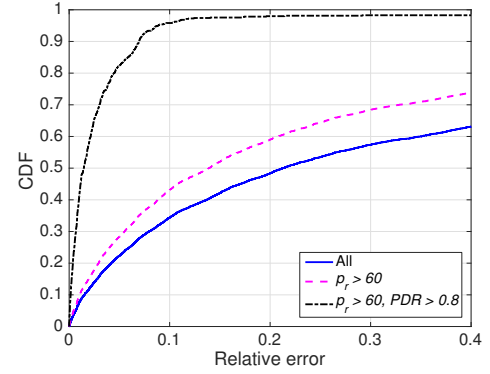


Fig. 12: Accuracy of surrogate model using Kriging method for different constraints in the full design space.

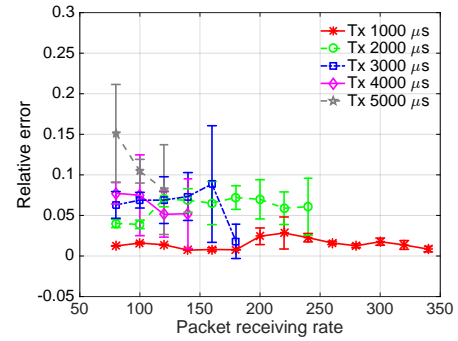


Fig. 13: Performance for the full design space with different packet receiving rate and average transmission time for packet receiving rate larger than 60 and PDR larger than 0.8.

point with average transmission time T_x , consists of different coverage and packet size ranges than the ones used during the training.

The 6000 test data points used in previous section are again used in the simulation. However, unlike previous simulations performed for the reduced design space, for a data point with average transmission time T_x , the simulator randomly selects a coverage and packet size range which can lead to the same T_x . For example, for T_x of 1000 μs , there are 7640 different cover-

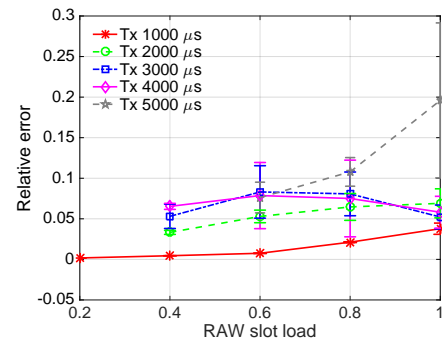


Fig. 14: Performance for the full design space with different slot load ratio and average transmission time for packet receiving rate larger than 60 and PDR larger than 0.8.

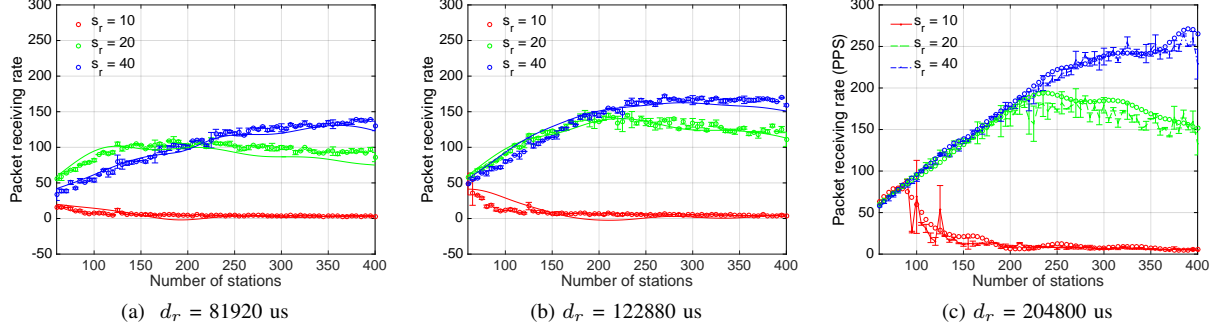


Fig. 15: Comparison between model and simulations results with fixed average transmission time (2000 us) under different stations number, slot number and RAW duration.

age and packet size range combinations ($[d^-, d^+, ps^-, ps^+]$), such as $[0, 80, 32, 256]$ and $[150, 170, 64, 128]$, etc. The simulation results are compared to the output of the built surrogate model, as shown in Figure 12. The results reveal that, in comparison to the reduced design space, the model accuracy is decreased. The lower model accuracy for the full design space is not surprising. In the modeling process, for each average transmission time, the network is configured with a fixed distance range and packet size. While in the full design space, for the same average transmission time, the network could be configured with many other coverage and packet size range combinations. Therefore, model accuracy is sacrificed in order to speed up the training process. However, as discussed in Section III-C, the model is mainly used for optimization, we are more interested in network configurations yielding large values of packet receiving rate and high PDR. By only taking into account the data points with packet receiving rate larger than 60 and PDR larger than 0.8, the model accuracy for the full design space is significantly improved. Around 95% of the data points have a relative error less than 0.1.

To further evaluate the model accuracy, we randomly choose 500 design points for each load ratio (i.e., 0.4, 0.6, 0.8, 1.0), consisting of 5 different average transmission times (i.e., 1000, 2000, 3000, 4000 and 5000 μs). During simulation, for each data point, the network is configured with a random coverage and packet size range, while satisfying the average transmission time T_x . Each simulation runs 10 times. For the evaluation, we only take into account the data points with packet receiving rate larger than 60 and PDR larger than 0.8. The relative error for each data point between the simulation results and the model prediction is calculated. For each average transmission time, the relative error is averaged among data points with the same packet receiving rate and slot load ratio, respectively. Furthermore, these two relative errors are averaged among all iterations, and depicted in Figures 13 and 14, respectively.

The results show, for the full design space, the built surrogate model is able to accurately predict the network performance. As depicted in Figure 13, the relative error is between 0.018 ± 0.021 and 0.103 ± 0.002 for most data points. Similar results are obtained in Figure 14. For a RAW slot load ratio no more than 0.4, the relative error is between

0.002 ± 0.0001 and 0.065 ± 0.004 . With a larger RAW slot ratio, for average transmission time between 1000 and 4000 μs , the relative error is between 0.052 ± 0.008 and 0.075 ± 0.047 , and between 0.076 ± 0.018 and 0.196 ± 0.095 for average transmission time 5000 μs .

D. A practical example

In this section, we evaluate the model accuracy for a practical IoT network scenario. As an example, we evaluate a scenario in which stations are randomly located around the AP with a distance no more than 500 meters, and have packet sizes between 64 and 512 bytes. The average transmission time of all stations is around 2000 μs . Figure 15 compares the surrogate model and simulation results under different number of stations, numbers of slots and RAW durations. The dots represent the simulation results, and the lines represent the model prediction. As it shows, the simulation results and the model predictions are quite close in most cases, and most importantly, always follow the same trend. Moreover, there are two discoveries on the RAW performance. First, the results show that a large number of RAW slots often leads to better performance, especially for large-scale networks. Second, a large RAW duration normally leads to higher packet rate, which is straightforward as there is more airtime for packet transmission. While too long RAW duration is a waste of airtime, as stations not belonging to the RAW group cannot access the channel. The appropriate RAW duration should be just long enough for transmitting all the packets.

V. CONCLUSION AND FUTURE WORK

In this paper, we present a new RAW performance model, based on supervised surrogate modeling, for IEEE 802.11ah heterogeneous networks, consisting of two novel contributions.

First, to the best of our knowledge, we are the first to present a RAW model for heterogeneous networks. The model can accurately predict the packet receiving rate (i.e., number of packets received per second) under a given RAW configuration in IEEE 802.11ah heterogeneous networks, in which stations have different packet sizes and MCSs based on their distance to the AP. Second, the training methodology is well designed to properly choose the input and output parameters of the

model, allowing the model to be trained with only a very limited number of sample data points. Realistic simulation results are compared to the prediction of the built surrogate model. The results show, although the training only uses one out of 5.7×10^{13} of all possible data points, the model can accurately predict the performance with a relative error less than 0.1 for 95% of the data points in “well-behaved” networks (i.e. suitable p_r and PDR values).

In future work, we aim to further extend the surrogate modeling approach to support other metrics (e.g., energy consumption), and approximate the optimal pareto front for multiple conflicting objectives. Moreover, we will investigate real-time RAW optimization methods for IEEE 802.11ah heterogeneous networks, in order to dynamically adapt the RAW configuration based on the networks conditions, using the developed model.

ACKNOWLEDGMENTS

Part of this research was funded by the Flemish FWO SBO S004017N IDEAL-IoT (Intelligent DENSE And Long range IoT networks) project, and by the ERDF and the Spanish Government through project TEC2016-79988-P, AEI/FEDER, UE.

REFERENCES

- [1] “IEEE standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan mac and phy specifications amendment 2: Sub 1 ghz license exempt operation,” *IEEE Std 802.11ah-2016 (Amendment to IEEE Std 802.11-2016, as amended by IEEE Std 802.11ai-2016)*, pp. 1–594, April 2017.
- [2] L. Tian, J. Famaey, and S. Latré, “Evaluation of the IEEE 802.11ah restricted access window mechanism for dense IoT networks,” in *IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016.
- [3] Y. Wang, Y. Li, K. K. Chai, Y. Chen, and J. Schormans, “Energy-aware adaptive restricted access window for IEEE 802.11ah based smart grid networks,” in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2015, pp. 581–586.
- [4] O. Raeesi, J. Pirskanen, A. Hazmi, T. Levanen, and M. Valkama, “Performance evaluation of IEEE 802.11ah and its restricted access window mechanism,” in *IEEE International Conference on Communications Workshops (ICC)*, 2014, pp. 460–466.
- [5] E. Khorov, A. Krotov, and A. Lyakhov, “Modelling machine type communication in IEEE 802.11ah networks,” *IEEE International Conference on Communication Workshop (ICCW)*, no. 14, pp. 1149–1154, 2015.
- [6] L. Zheng, M. Ni, L. Cai, J. Pan, C. Ghosh, and K. Doppler, “Performance analysis of group-synchronized dcf for dense IEEE 802.11 networks,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 11, pp. 6180–6192, 2014.
- [7] A. Lyakhov, E. Khorov, and R. Yusupov, “Two-slot based model of the IEEE 802.11ah restricted access window with enabled transmissions crossing slot boundaries,” in *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2018.
- [8] C. W. Park, D. Hwang, and T.-J. Lee, “Enhancement of IEEE 802.11ah MAC for M2M communications,” *IEEE Communications Letters*, vol. 18, no. 7, pp. 1151–1154, 2014.
- [9] L. Tian, E. Khorov, S. Latré, and J. Famaey, “Real-time station grouping under dynamic traffic for IEEE 802.11ah,” *Sensors*, vol. 17, no. 7, 2017.
- [10] L. Tian, S. Santi, S. Latré, and J. Famaey, “Accurate sensor traffic estimation for station grouping in highly dense IEEE 802.11ah networks,” in *15th ACM Conference on Embedded Networked Sensor Systems Workshops (SenSys)*, 2017.
- [11] L. Tian, M. Mehari, S. Santi, S. Latré, E. De Poorter, and J. Famaey, “IEEE 802.11ah restricted access window surrogate model for real-time station grouping,” in *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2018.
- [12] L. Tian, M. Mehari, S. Santi, S. Latré, E. D. Poorter, and J. Famaey, “Multi-objective surrogate modeling for real-time energy-efficient station grouping in IEEE 802.11ah,” *Pervasive and Mobile Computing*, 2019.
- [13] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, “A survey on IEEE 802.11ah: An enabling networking technology for smart cities,” *Computer Communications*, vol. 58, pp. 53–69, 2015.
- [14] V. Baos-Gonzalez, M. S. Afaqui, E. Lopez-Aguilera, and E. Garcia-Villegas, “IEEE 802.11ah: A technology to face the IoT challenge,” *Sensors*, vol. 16, no. 11, 2016.
- [15] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, “A surrogate modeling and adaptive sampling toolbox for computer based design,” *J. Mach. Learn. Res.*, vol. 11, pp. 2051–2055, Aug. 2010.
- [16] M. Mehari, E. De Poorter, I. Couckuyt, D. Deschrijver, G. Vermeeren, D. Plets, W. Joseph, L. Martens, T. Dhaene, and I. Moerman, “Efficient identification of a multi-objective pareto front on a wireless experimentation facility,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 6662–6675, 2016.
- [17] M. T. Mehari, E. De Poorter, I. Couckuyt, D. Deschrijver, J. Vanhie-Van Gerwen, D. Pareit, T. Dhaene, and I. Moerman, “Efficient global optimization of multi-parameter network problems on wireless testbeds,” *Ad Hoc Networks*, vol. 29, pp. 15–31, 2015.
- [18] H. Braham, S. B. Jemaa, G. Fort, E. Moulines, and B. Sayrac, “Fixed rank kriging for cellular coverage analysis,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4212–4222, May 2017.
- [19] L. Tian, S. Deronne, S. Latré, and J. Famaey, “Implementation and validation of an IEEE 802.11ah module for ns-3,” in *Proceedings of the Workshop on Ns-3 (WNS3)*, 2016, pp. 49–56.
- [20] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000.
- [21] T. Chang, C. Lin, K. C. Lin, and W. Chen, “Traffic-aware sensor grouping for IEEE 802.11ah networks: Regression based analysis and design,” *IEEE Transactions on Mobile Computing*, pp. 1–1, 2018.
- [22] A. Ba, Y.-H. Liu, and e. a. van den Heuvel, Johan, “26.3A 1.3nJ/b IEEE 802.11ah fully digital polar transmitter for IoT applications,” in *IEEE International Solid-State Circuits Conference*, 2016, pp. 440–441.
- [23] B. Bellekens, L. Tian, P. Boer, M. Weyn, and J. Famaey, “Outdoor IEEE 802.11ah range characterization using validated propagation models,” in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, IEEE, 2017, pp. 1–6.
- [24] A. Kamerman and L. Monteban, “Wavelan®-ii: a high-performance wireless LAN for the unlicensed band,” *Bell Labs technical journal*, vol. 2, no. 3, pp. 118–133, 1997.
- [25] M. Lacage, M. H. Manshaei, and T. Turletti, “IEEE 802.11 rate adaptation: a practical approach,” in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*. ACM, 2004, pp. 126–134.
- [26] Madwifi project. [Online]. Available: <http://madwifi.sourceforge.net/>
- [27] Minstrel rate adaptation algorithm documentation. [Online]. Available: https://sourceforge.net/p/madwifi/svn/HEAD/tree/madwifi/trunk/ath_rate/minstrel/minstrel.txt
- [28] F. Viana, “Things you wanted to know about the latin hypercube design and were afraid to ask,” in *the 10th World Congress on Structural and Multidisciplinary Optimization*, 2013.
- [29] A. Forrester, A. Keane et al., *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.
- [30] J. R. Edwards, “Polynomial regression and response surface methodology,” *Perspectives on organizational fit*, pp. 361–372, 2007.
- [31] M. D. Buhmann, *Radial basis functions: theory and implementations*. Cambridge university press, 2003, vol. 12.
- [32] B. Schölkopf, A. J. Smola, F. Bach et al., *Learning with kernels: support vector machines, regularization, optimization*. MIT press, 2002.
- [33] J. W. Bandler, Q. S. Cheng, S. A. Dakrouy, A. S. Mohamed, M. H. Bakr, K. Madsen, and J. Sondergaard, “Space mapping: the state of the art,” *IEEE Transactions on Microwave theory and techniques*, vol. 52, no. 1, pp. 337–361, 2004.
- [34] A. K. Jain, J. Mao, and K. Mohiuddin, “Artificial neural networks: A tutorial,” *Computer*, no. 3, pp. 31–44, 1996.
- [35] J. van der Herten, I. Couckuyt, D. Deschrijver, and T. Dhaene, “A fuzzy hybrid sequential design strategy for global surrogate modeling of high-dimensional computer experiments,” *SIAM Journal of Scientific Computing*, vol. 37, no. 2, pp. 1020–1039, 2015.
- [36] L. Tian, A. Šljivo, S. Santi, E. De Poorter, J. Hoebeke, and J. Famaey, “Extension of the IEEE 802.11ah ns-3 simulation module,” in *Proceedings of the 10th Workshop on Ns-3, ser. WNS3 ’18*. New York, NY, USA: ACM, 2018, pp. 53–60.